PUBLIC KEY CRYPTOGRAPHIC APPARATUS AND METHOD

Sub B1 a1>

~~This application claims the benefit of U.S. Provisional~~ Application No. 60/033,271 for PUBLIC KEY CRYTOGRAPHIC APPARATUS AND METHOD, filed Dec. 9, 1996, naming as inventors, Thomas Colins, Dale Hopkins, Susan Langford and Michale Sabin, the discolsure of which ~~is incorporated by reference.~~

BACKGROUND OF THE INVENTION

This invention relates generally to communicating data in a secure fashion, and more particularly to a cryptographic system and methods using public key cryptography.

Computer systems are found today in virtually every walk of life for storing, maintaining, and transferring various types of data. The integrity of large portions of this data, especially that portion relating to financial transactions, is vital to the health and survival of numerous commercial enterprises. Indeed, as open and unsecured data communications channels for sales transactions gain popularity, such as credit card transactions over the Internet, individual consumers have an increasing stake in data security.

Thus, for obvious reasons, it is important that financial transaction communications pass from a sender to an intended receiver without intermediate parties being able to interpret the transferred message.

Cryptography, especially public key cryptography, has proven to be an effective and convenient technique of enhancing data privacy and authentication. Data to be secured, called plaintext, is transformed into encrypted data, or ciphertext, by a predetermined encryption process of one type or another. The reverse process, transforming ciphertext into plaintext, is termed decryption. Of particular importance to this invention is that the processes of encryption and decryption are controlled by a pair of related cryptographic keys. A "public" key is used for the encryption process, and a "private" key is used to decrypt ciphertext. The public key transforms plaintext to ciphertext, but cannot be used to decrypt the ciphertext to retrieve the plaintext therefrom.

As an example, suppose a Sender A wishes to send message M to a recipient B. The idea is to use public key E and related private key D for encryption and decryption of M. The public key E is public information while D is kept secret by the intended receiver. Further, and importantly, although E is determined by D, it is extremely difficult to compute D from E. Thus the receiver, by publishing the public key E, but keeping the private key D secret, can assure senders of data encrypted using E that anyone who intercepts the data will not be able to decipher it. Examples of the public key/private key concept can be found in U.S. Pat. Nos. 4,200,770, 4,218,582, and 4,424,414.

The prior art includes a number of public key schemes, in addition to those described in the above-identified patents. Over the past decade, however, one system of public key cryptography has gained popularity. Known generally as the "RSA" scheme, it is now thought by many to be a worldwide defacto standard for public key cryptography. The RSA scheme is described in U.S. Pat. No. 4,405,829 which is fully incorporated herein by this reference.

Sub B2 a2>

~~The RSA scheme capitalizes on the relative ease of~~ creating a composite number from the product of two prime numbers whereas the attempt to factor the composite number into its constituent primes is difficult. The RSA scheme uses a public key E comprising a pair of positive integers n and e. where n is a composite number of the form

$$n=p\cdot q \tag{1}$$

where p and q are different prime numbers. and e is a number relatively prime to (p−1) and (q−1): that is. e is relatively prime to (p−1) or (q−1) if e has no factors in common with either of them. Importantly. the sender has access to n and e. but not to p and q. The message M is a number representative of a message to be transmitted wherein

$$0\leqq M<n-1. \tag{2}$$

The sender enciphers M to create ciphertext C by computing the exponential

~~$C=M^e(\text{mod } n)$~~ (3)

Sub B3 a3

~~The recipient of the ciphertext C retrieves the message M~~ using a (private) decoding key D. comprising a pair of positive integers d and n. employing the relation

$$M=C^d(\text{mod } n) \tag{4}$$

As used in (4). above. d is a multiplicative inverse of

$$e(\text{mod}(\text{lcm}((p-1).\ (q-1)))) \tag{5}$$

so that

$$e\cdot d=1(\text{mod}(\text{lcm}((p-1).\ (q-1)))) \tag{6}$$

where lcm((p−1). (q−1)) is the least common multiple of numbers p−1 and q−1. Most commercial implementations of RSA employ a different. although equivalent. relationship for obtaining d:

$$d=e^{-1}\text{mod}(p-1)\ (q-1). \tag{7}$$

~~This alternate relationship simplifies computer processing.~~

Note: Mathematically (6) defines a set of numbers and (7) defines a subset of that set. For implementation. (7) or (6) usually is interpreted to mean d is the smallest positive element in the set.)

The net effect is that the plaintext message M is encoded knowing only the public key E (i.e.. e and n). The resultant ciphertext C can only decoded using decoding key D. The composite number n. which is part of the public key E. is computationally difficult to factor into its components. prime numbers p and q. a knowledge of which is required to decrypt C.

From the time a security scheme. such as RSA. becomes publicly known and used. it is subjected to unrelenting attempts to break it. One defense is to increase the length (i.e.. size) of both p and q. Not long ago it was commonly recommended that p and q should be large prime numbers 75 digits long (i.e.. on the order of 10^{75}). Today. it is not uncommon to find RSA schemes being proposed wherein the prime numbers p and q are on the order of 150 digits long. This makes the product of p and q a 300 digit number. (There are even a handful of schemes that employ prime numbers (p and q) that are larger. for example 300 digits long to form a 600 digit product.) Numbers of this size. however. tend to require enormous computer resources to perform the encryption and decryption operations. Consider that while computer instruction cycles are typically measured in nanoseconds (billionths of seconds). computer computations of RSA steps are typically measured in milliseconds (thousandths of seconds). Thus millions of computer cycles are required to compute individual RSA steps resulting in noticeable delays to users.

This problem is exacerbated if the volume of ciphertext messages requiring decryption is large—such as can be expected by commercial transactions employing a mass communication medium such as the Internet. A financial institution may maintain an Internet site that could conceivably receive thousands of enciphered messages every hour that must be decrypted. and perhaps even responded to. Using larger numbers to form the keys used for an RSA scheme can impose severe limitations and restraints upon the institution's ability to timely respond.

Many prior art techniques. while enabling the RSA scheme to utilize computers more efficiently. nonetheless have failed to keep pace with the increasing length of n. p. and q.

Accordingly. it is an object of this invention to provide a system and method for rapid encryption and decryption of data without compromising data security.

It is another object of this invention to provide a system and method that increases the computational speed of RSA encryption and decryption techniques.

sub B4 ~~a4~~

~~It is still another object of this invention to provide a system and method for implementing an RSA scheme in which the components of n do not increase in length as n increases in length.~~

Sub ~~a5~~ B5

~~It is still another object to provide a system and method for utilizing multiple (more than two). distinct prime number components to create n.~~

It is a further object to provide a system and method for providing a technique for reducing the computational effort for calculating exponentiations in an RSA scheme for a given length of n.

SUMMARY OF THE INVENTION

Sub B6 ~~a6~~

~~The present invention discloses a method and apparatus for increasing the computational speed of RSA and related public key schemes by focusing on a neglected area of computation inefficiency. Instead of n=p·q. as is universal in the prior art. the present invention discloses a method and apparatus wherein n is developed from three or more distinct prime numbers: i.e.. $n=p_1 \cdot p_2 \cdot \ldots \cdot p_k$. where k is an integer greater than 2 and $p_1, p_2, \ldots p_k$ are sufficiently large distinct primes. Preferably. "sufficiently large primes" are prime numbers that are numbers approximately 150 digits long or larger. The advantages of the invention over the prior art should be immediately apparent to those skilled in this art. If. as in the prior art. p and q are each on the order of. say. 150 digits long. then n will be on the order of 300 digits long. However. three primes p_2. p_1. and p_3 employed in accordance with the present invention can each be on the order of 100 digits long and still result in n being 300 digits long. Finding and verifying 3 distinct primes. each 100 digits long. requires significantly fewer computational cycles than finding and verifying 2 primes each 150 digits long.~~

Sub ~~a7~~ B7

~~The commercial need for longer and longer primes shows no evidence of slowing: already there are projected requirements for n of about 600 digits long to forestall incremental improvements in factoring techniques and the ever faster computers available to break ciphertext. The invention. allowing 4 primes each about 150 digits long to obtain a 600 digit n. instead of two primes about 350 digits long. results in a marked improvement in computer performance. For. not only are primes that are 150 digits in size easier to find and verify than ones on the order of 350 digits. but by applying techniques the inventors derive from the Chinese Remainder Theorem (CRT). public key cryptography calculations for~~

encryption and decryption are completed ~~much~~ faster—even if performed serially on a ~~single~~ processor system. However, the inventors' ~~techniques~~ are particularly adapted to be advantageously apply enable public key operations to par~~allel computer processing.~~

Sub B8 ~~a8~~

~~The present invention is capable of using the RSA scheme~~ to perform encryption and decryption ~~operation~~ using a large (many digit) n much faster than heretofore possible. Other advantages of the invention include its employment for decryption without the need to revise the RSA public encryption transformation scheme currently in use on thou~~sands of large and small computers.~~

Sub ~~a9~~ B9

~~A key assumption of the present invention is that n.~~ composed of 3 or more sufficiently large distinct prime numbers, is no easier (or not very much easier) to factor than the prior art, two prime number n. The assumption is based on the observation that there is no indication in the prior art literature that it is "easy" to factor a product consisting of more than two sufficiently large, distinct prime numbers. This assumption may be justified given the continued effort (and failure) among experts to find a way "easily" to break large component numbers into their large prime factors. This assumption is similar, in the inventors' view, to the assumption underlying the entire field of public key cryptography that factoring composite numbers made up of two distinct primes is not "easy." That is, the entire field of public key cryptography is based not on mathematical proof, but on the assumption that the empirical evidence of failed sustained efforts to find a way systematically to solve NP problems in polynomial time indicates that these problems truly are ~~"difficult."~~

Sub ~~a10~~ B10

~~The invention is preferably implemented in a system that~~ employs parallel operations to perform the encryption, decryption operations required by the RSA scheme. Thus, there is also disclosed a cryptosystem that includes a central processor unit (CPU) coupled to a number of exponentiator elements. The exponentiator elements are special purpose arithmetic units designed and structured to be provided message data M, an encryption key e, and a number n (where $n=p_1*p_2*\ldots p_k$, k being greater than 2) and return ciphertext C according to the relationship.

~~$C=M^e(\mathrm{mod}(n))$.~~

Sub ~~a11~~ B11

~~Alternatively, the exponentiator elements may be pro~~vided the ciphertext C, a decryption (private) key d and n to return M according to the relationship.

~~$M=C^d(\mathrm{mod}(n))$.~~

Sub B12 ~~a12~~

~~According to this aspect of the invention, the CPU~~ receives a task, such as the requirement to decrypt cypher-text data C. The CPU will also be provided, or have available, a public key e and n, and the factors of n ($p_1, p_2, \ldots p_k$). The CPU breaks the encryption task down into a number of sub-tasks, and delivers the sub-tasks to the exponentiator elements. When the results of the sub-tasks are returned by the exponentiator elements to the CPU which will, using a form of the CRT, combine the results to obtain the message data M. An encryption task may be performed essentially in the same manner by the CPU and its use of the exponentiator elements. However, usually the factors of n are not available to the sender (encryptor), only ~~the public key, e and n, so that no sub-tasks are created.~~

In a preferred embodiment of this latter aspect of the invention, the bus structure used to couple the CPU and exponentiator elements to one another is made secure by encrypting all important information communicated thereon. Thus. data sent to the exponentiator elements is passed through a data encryption unit that employs. preferably. the ANSI Data Encryption Standard (DES). The exponentiator elements decrypt the DES-encrypted sub-task information they receive. perform the desired task. and encrypt the result. again using DES. for return to the CPU.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a simplified block diagram of a cryptosystem architecture configured for use in the present invention.

FIG. 2 is a memory map of the address space of the cryptosystem of FIG. 1; and

FIG. 3 is an exemplary illustration of one use of the invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

As indicated above. the present invention is employed in the context of the RSA public key encryption/decryption scheme. As also indicated. the RSA scheme obtains its security from the difficulty of factoring large numbers. and the fact that the public and private keys are functions of a pair of large (100–200 digits or even larger) prime numbers. Recovering the plaintext from the public key and the ciphertext is conjectured to be equivalent to factoring the product of two primes.

Sub B14 ~~a14~~

~~According to the present invention. the public key portion~~ e is picked. Then. three or more random large. distinct prime numbers. $p_1, p_2, \ldots, p_k$ are developed and checked to ensure that each is relatively prime to e. Preferably. the prime numbers are of equal length. Then. the product $n=p_1 \cdot p_2 \cdot \ldots \cdot$ ~~p_k is computed.~~

Sub a15

~~Finally. the decryption key. d. is established by the~~ relationship:

~~$d=e^{-1} \bmod ((p_1-1)(p_2-1) \ldots (p_k-1))$~~

Sub B15 ~~a16~~

~~The message data. M is encrypted to ciphertext C using~~ the relationship of (3). above. i.e..

~~$C=M^e \bmod n.$~~

Sub B16 ~~a17~~

~~To decrypt the ciphertext. C. the relationship of (3).~~ above. is used:

~~$M=C^d \bmod n$~~

~~where n and d are those values identified above.~~

Ins. new ¶ ~~a13~~ B13

a13

Sub B17 ~~a18~~

~~Using the present invention involving three primes to~~ develop the product n. RSA encryption and decryption time can be substantially less than an RSA scheme using two primes by dividing the encryption or decryption task into sub-tasks. one sub-task for each distinct prime. (However. breaking the encryption or decryption into subtasks requires knowledge of the factors of n. This knowledge is not usually available to anyone except the owner of the key. so the encryption process can be accelerated only in special cases. such as encryption for local storage. A system encrypting data for another user performs the encryption process according to (3). independent of the number of factors of n. Decryption. on the other hand. is performed by the owner of a key. so the factors of n are generally known and can be used to accelerate the process.) For example. assume that three distinct primes. p_1, p_2, and p_3. are used to develop the product n. Thus. decryption of the ciphertext. C. using the relationship

$$M=C^d(\mathrm{mod}\ n)$$

is used to develop the decryption sub-tasks:

$$M_1=C_1^{d_1}\mathrm{mod}\ p_1$$

$$M_2=C_2^{d_2}\mathrm{mod}\ p_2$$

$$M_3=C_3^{d_3}\mathrm{mod}\ p_3$$

where

$$C_1=C\mathrm{mod}\ p_1;$$

$$C_2=C\mathrm{mod}\ p_2;$$

$$C_3=C\mathrm{mod}\ p_3;$$

$$d_1=d\mathrm{mod}\ (p_1-1);$$

$$d_2=d\mathrm{mod}\ (p_2-1);\text{ and}$$

~~$d_3=d\mathrm{mod}\ (p_3-1);$~~

Sub ~~a19~~ B18

~~The results of each sub-task. M_1. M_2. and M_3 can be~~ combined to produce the plaintext. M. by a number of techniques. However. it is found that they can most expeditiously be combined by a form of the Chinese Remainder Theorem (CRT) using. preferably. a recursive scheme. Generally. the plaintext M is obtained from the combination of the individual sub-tasks by the following relationship:

$$Y_i=Y_{i-1}+[(M_i-Y_{i-1})\ (w_i^{-1}\mathrm{mod}\ p_i)\mathrm{mod}\ p_i]\cdot w_i\mathrm{mod}\ n$$

where

$$i\geqq 2\text{ and}$$

$$M=Y_k,\ Y_1=C_1,\text{ and } w_i=\prod_{j<i} p_j$$

Encryption is performed in much the same manner as that used to obtain the plaintext M. provided (as noted above) the factors of n are available. Thus. the relationship

$$C=M^e(\mathrm{mod}\ n),$$

can be broken down into the three sub-tasks.

$$C_1=M_1^{e_1}\mathrm{mod}\ p_1$$

$$C_2=M_2^{e_2}\mathrm{mod}\ p_2$$

$$C_3=M_3^{e_3}\mathrm{mod}\ p_3$$

where

$$M_1=M(\mathrm{mod}\ p_1),$$

$$M_2=M(\mathrm{mod}\ p_2),$$

$$M_3=M(\mathrm{mod}\ p_3),$$

$$e_1=e\mathrm{mod}\ (p_1-1),$$

$$e_2=e\mathrm{mod}\ (p_2-1),\text{and}$$

~~$e_3=e\mathrm{mod}\ (p_3-1).$~~

Sub ~~a20~~ B19

~~In generalized form. the decrypted message M can be~~ obtained by the ~~same~~ summation identified above to obtain ~~the ciphertext C from its contiguous constituent sub-tasks C_i.~~

Sub ~~a21~~ B20

~~Preferably. the recursive CRT method described above is~~ used to obtain either the ciphertext. C. or the deciphered plaintext (message) M due to its speed. However. there may be occasions when it is beneficial to use a non-recursive technique in which case the following relationships are used:

$$M = \sum_{i=1}^{k} M_i(w_i^{-1} \, mod \, p_i)w_i \, mod \, n$$

where

$$w_i = \prod_{j \neq i} p_j, \text{ and}$$

k is the number (3 or more) of distinct primes chosen to ~~develop the product n.~~

Sub ~~a22~~ B21

~~Thus. for example above (k=3). M is construated from the~~ returned sub-task values M_1. M_2. M_3 by the relationship

$$M=M_1(w_1^{-1}\text{mod } p_1)\ w_1\text{mod } n+M_2(w_2^{-1}\text{mod } p_2)\ w_2\text{mod } n\ +M_3(w_3^{-1}\text{mod } p_3)\ w_3\text{mod } n$$

where

~~$w_1=p_2p_3$. $w_2=p_1p_3$. and $w_3=p_1p_2$.~~

Employing the multiple distinct prime number technique of the present invention in the RSA scheme can realize accelerated processing over that using only two primes for the same size n. The invention can be implemented on a single processor unit or even the architecture disclosed in the above-referenced U.S. Pat. No. 4.405.829. The capability of developing sub-tasks for each prime number is particularly adapted to employing a parallel architecture such as that illustrated in FIG. 1.

Turning to FIG. 1. there is illustrated a cryptosystem architecture apparatus capable of taking particular advantage of the present invention. The cryptosystem. designated with the reference numeral **10**. is structured to form a part of a larger processing system (not shown) that would deliver to the cryptosystem **10** encryption and/or decryption requests. receiving in return the object of the request—an encrypted or decrypted value. The host would include a bus structure **12**. such as a peripheral component interface (PCI) bus for communicating with the cryptosystem **10**.

As FIG. **1** shows. The cryptoprocessor **10** includes a central processor unit (CPU) **14** that connects to the bus structure **12** by a bus interface **16**. The CPU **14** comprises a processor element **20**. a memory unit **22**. and a data encryption standard (DES) unit **24** interconnected by a data/address bus **26**. The DES unit **24**. in turn. connects to an input/output (I/O) bus **30** (through appropriate driver/receiver circuits—not shown).

Sub ~~a23~~ B22

~~The I/O bus 30 communicatively connects the CPU to a~~ number of exponentiator elements 32_a. 32_b. and 32_c. Shown here are three exponentiator elements. although as illustrated by the "other" exponentiators 32_n. additional exponentiator elements can be added. Each exponentiator element is a state machine controlled arithmetic circuit structured specifically to implement the relationship described above. Thus. for example. the exponentiator **32***a* would be provided the values M_1. e_1. and p. n to develop C_1. Similarly. the exponentiator circuits **32***b* and **32***c* develop C_2 and C_3 from ~~corresponding subtask values M_2. e_2. P_2. M_3. e_3. and P_3.~~

Preferably. the CPU **14** is formed on a single integrated circuit for security reasons. However. should there be a need for more storage space than can be provided by the "on-board" memory **22**. the bus **30** may also connect the CPU **14** to an external memory unit **34**.

Sub
a24
B23

~~In order to ensure a secure environment, it is preferable~~ that the cryptosystem **10** meet the Federal Information Protection System (FIPS) level 3. Accordingly. the elements that make up the CPU **14** would be implemented in a design that will be secure from external probing of the circuit. However. information communicated on the I/O bus **30** between the CPU **14** and the exponentiator circuits **32** (and external memory **34**—if present) is exposed. Consequently. to maintain the security of that information. it is first encrypted by the DES unit **24** before it is placed on the I/O bus **30** by the CPU **14**. The exponentiator circuits **32**. as well as the external memory **34**. will also include similar DES units to decrypt information received from the CPU. and ~~later to encrypt information returned to the CPU 14.~~

It may be that not all information communicated on the I/O bus **30** need be secure by DES encryption. For that reason. the DES unit **24** of the CPU **14** is structured to encrypt outgoing information. and decrypt incoming information. on the basis of where in the address space used by the cryptosystem the information belongs; that is. since information communicated on the I/O bus **30** is either a write operation by the CPU **14** to the memory **34**. or a read operation of those elements. the addresses assigned to the secure addresses and non-secure addresses. Read or write operations conducted by the CPU **14** using secure addresses will pass through the DES unit **24** and that of the memory **34**. Read or write operations involving non-secure addresses will by-pass these DES units.

FIG. 2 diagrammatically illustrates a memory map **40** of the address space of the cryptosystem **10** that is addressable by the processor **20**. As the memory map **40** shows. an address range **40** provides addresses for the memory **22**. and such other support circuitry (e.g.. registers—not shown) that may form a part of the CPU **14**. The addresses used to write information to. or read information from. the exponentiator elements **32** are in the address range **44** of the memory map **40**. The addresses for the external memory **34** are in the address ranges **46**. and **48**. The address ranges **44** and **46** are for secure read and write operations. Information that must be kept secure. such as instructions for implementing algorithms. encryption/decryption keys. and the like. if maintained in external memory **34**. will be stored at locations having addresses in the address range **46**. Information that need not be secure such as miscellaneous algorithms data. general purpose instructions. etc. are kept in memory locations of the external memory **34** having addresses within the address range **48**.

The DES unit **24** is structured to recognize addresses in the memory spaces **44**. **46**. and to automatically encrypt the information before it is applied to the I/O bus **30**. The DES unit **24** is bypassed when the processor **20** accesses addresses in the address range **48**. Thus. when the processor **20** initiates write operations to addresses within the memory space within the address range **46** (to the external memory **34**). the DES unit **24** will automatically encrypt the information (not the addresses) and place the encrypted information on the I/O bus **30**. Conversely. when the processor **20** reads information from the external memory **34** at addresses within the address range **46** of the external memory **34**. the DES unit will decrypt information received from the I/O bus **30** and place the decrypted information on the data/address bus **26** for the processor **20**.

Sub
B24 a25

~~In similar fashion. information conveyed to or retrieved~~ from the exponentiators **32** by the processor **20** by write or read operations at addresses within the address range **44**. Consequently. writes to the exponentiators **32** will use the DES unit **24** to encrypt the information. When that (encrypted) information is received by the exponentiators

32. it is decrypted by on-board DES units ~~(of each exponentiator 32). The results of the task performed by the exponentiator 32 is then encrypted by the exponentiator's on-board DES unit. retrieved by the processor 20 in encrypted form and then decrypted by the DES unit 24.~~

Information that need not be maintained in secure fashion to be stored in the external memory **34**. however. need only be written to addresses in the address range **48**. The DES unit **24** recognizes writes to the address range **48**. and bypasses the encryption circuitry. passing the information. in unencrypted form. onto the I/O bus **30** for storing in the external memory **34**. Similarly. reads of the external memory **34** using addresses within the address range **48** are passed directly from the I/O bus **30** to the data/address bus **26** by the DES unit **24**.

In operation. the CPU **14** will receive from the host it serves (not shown). via the bus **12**. an encryption request. The encryption request will include the message data M to be encrypted and. perhaps. the encryption keys e and n (in the form of the primes p_1. p_2. . . . p_k). Alternatively. the keys may be kept by the CPU **14** in the memory **22**. In any event. the processor **20** will construct the encryption sub-tasks C_1. C_2. . . . . C_k for execution by the exponentiators **32**.

Sub ~~a26~~ B25

~~Assume. for the purpose of the remainder of this discussion. that the encryption/decryption tasks performed by the cryptosystem **10**. using the present invention. employs only three distinct primes. p_1. p_2. p_3. The processor **20** will develop the sub tasks identified above. using M. e. p_1 p_2. p_3 Thus. for example. if the exponentiator **32***a* were assigned the sub-task of developing C_1. the processor would develop the values M_1. e_1. and (p_1-1) and deliver units (write) these values. with n. to the exponentiator **32***a*. Similar values will be developed by the processor **20** for the sub-tasks that will be delivered to the exponentiators **32***b* and **32***c*.~~

In turn. the exponentiators **32** develop the values C_1. C_2. and C_3 which are returned to (retrieved by) the CPU **14**. The processor **20** will then combine the values C_1. C_2. and C_3 to form C. the ciphertext encryption of M. which is then returned to the host via the bus **12**.

The encryption. decryption techniques described hereinabove. and the use of the cryptosystem **10** (FIG. **1**) can find use in a number of diverse environments. Illustrated in FIG. **3** is one such environment. FIG. **3** shows a host system **50**. including the bus **12** connected to a plurality of cryptosystems **10** (**10***a*. **10***b*. . . . . **10***m*) structured as illustrated in FIG. **1**. and described above. In turn. the host system **50** connects to a communication medium **60** which could be. for example. an internet connection that is also used by a number of communicating stations **64**. For example. the host system **50** may be employed by a financial institution running a web site accessible. through the communication medium. by the stations **64**. Alternatively. the communication medium may be implemented by a local area network (LAN) or other type network. Use of the invention described herein is not limited to the particular environment in which it is used. and the illustration in FIG. **3** is not meant to limit in any way how the invention can be used.

As an example. the host system. as indicated. may receive encrypted communication from the stations **64**. via the communication medium **60**. Typically. the data of the communication will be encrypted using DES. and the DES key will be encrypted using a public key by the RSA scheme. preferably one that employs three or more distinct prime numbers for developing the public and private keys.

Continuing. the DES encrypted communication. including the DES key encrypted with the RSA scheme. would be